

INFORME

Proyecto: Juego de Buscaminas

Info 229 - Arquitectura de software



Integrantes

Eduardo Leal

Luis Olivares

Sebastian Pangué

Profesor

Matthieu Vernier

1. Introducción

El Buscaminas es un videojuego que apareció por primera vez en 1992 con el objetivo de familiarizar al usuario en el manejo del ratón, el juego consiste en despejar todas las casillas de una pantalla que no oculten una mina, si se despeja una casilla que contenga una mina, se pierde.

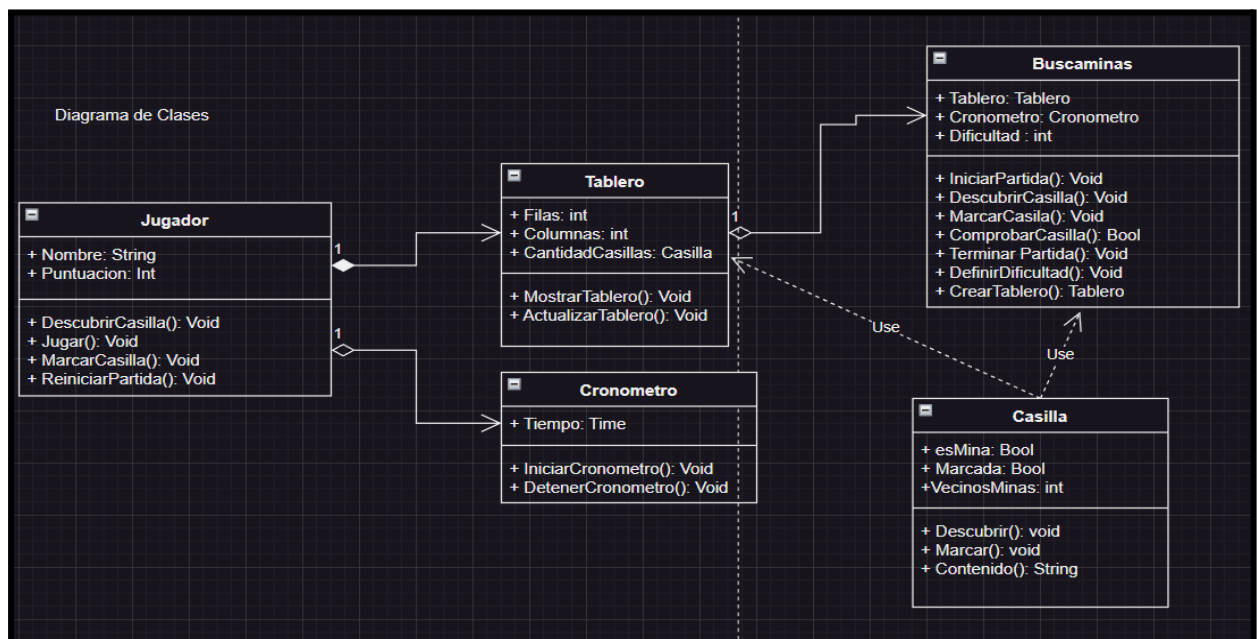
En este proyecto se busca diseñar e implementar nuestra propia versión del videojuego "Buscaminas" utilizando el lenguaje de programación "Python", además de hacer uso de la metodología 4+1, patrones de diseño y otras buenas prácticas para mejorar la calidad de este mismo.

2. Metodología de Diseño

La metodología 4+1 es un enfoque para diseñar arquitecturas de software que combina cuatro puntos de vista o perspectivas diferentes junto con un escenario de uso o "vista de casos de uso".

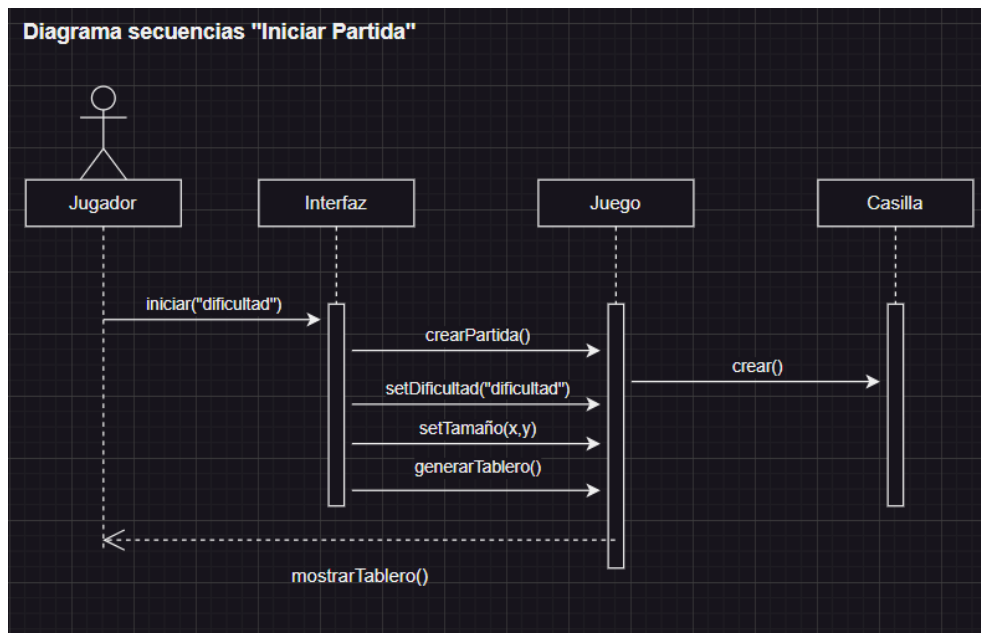
Nosotros previamente diseñamos cada diagrama relacionado a esta metodología, para posteriormente realizar el código en python, esto nos permitió mantener un orden y coherencia en la realización del software solicitado.

2.1 Vista Lógica (Diagrama de Clases):



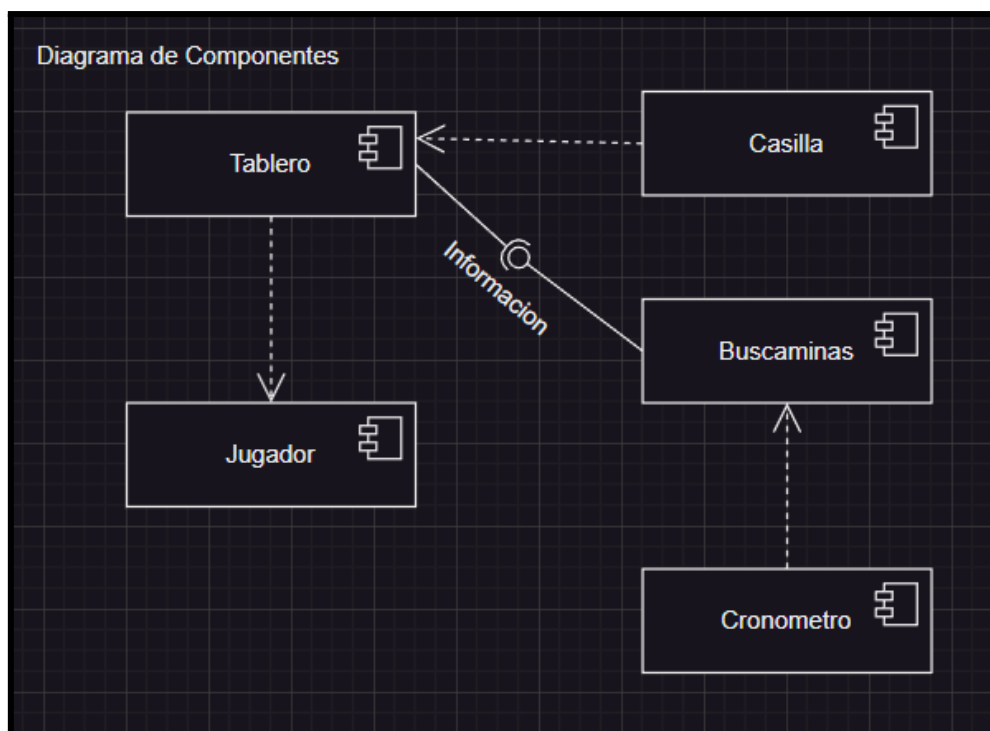
Muestra la funcionalidad y la relación entre los componentes más importantes vistos en la vista de desarrollo, y cómo se estructuran los elementos lógicos de cada uno.

2.2 Vista de Procesos (Diagrama de Secuencia):



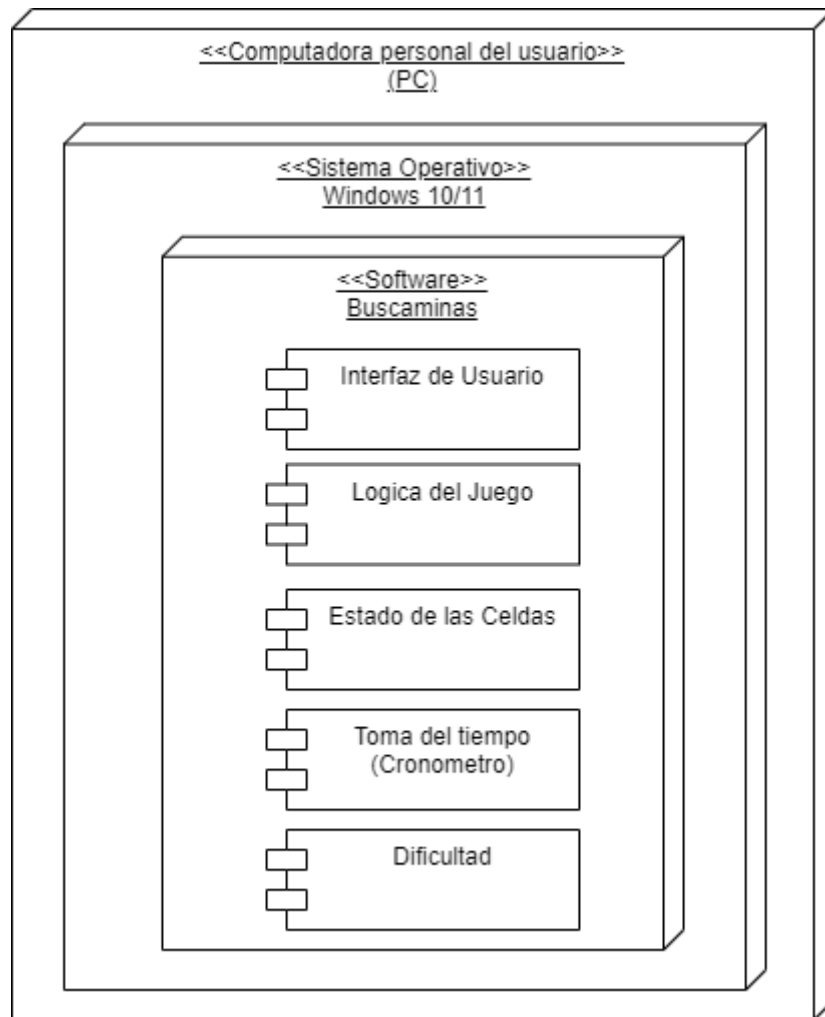
Demuestra el Comportamiento Dinámico en el proceso de “Iniciar Partida” y cómo los componentes interactúan entre sí en el momento de ejecución.

2.3 Vista de Desarrollo / Implementación (Diagrama de componentes):



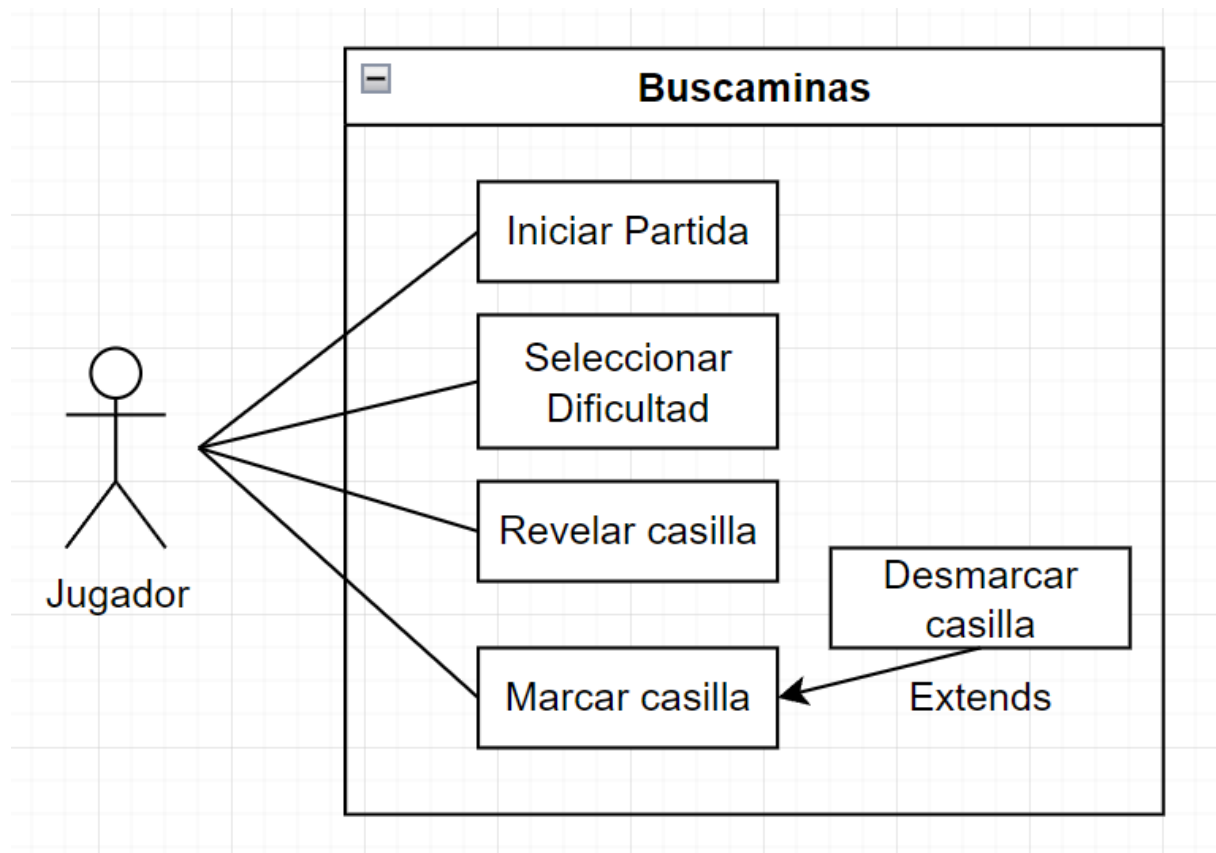
Aquí se muestra la Organización del software para el desarrollo, mostrando los módulos más importantes que serán desarrollados.

2.4 Vista Física (Diagrama de Despliegue):



Muestra cómo los componentes del Juego se despliegan en el entorno de hardware y software.

2.5 Casos de Uso (Diagrama de Casos de Uso):



Estos son los escenarios específicos que ilustran cómo el usuario “Jugador” interactúa con el sistema y cómo se realizan las operaciones de “Iniciar Partida”, “Descubrir Casilla” y “Marcar Casilla” que son las funciones más importantes del Juego.

4. Patrones de Diseño

Patron Observador:

Este patrón se utiliza cuando hay una relación de uno a muchos entre objetos, cuando un objeto cambia su estado, todos los dependientes son notificados y actualizados automáticamente. Ésta lógica es justamente lo que necesitamos para desarrollar nuestro buscaminas, el objeto principal serían las casillas individuales presentes en el tablero, y los objetos dependientes serían todos los demás componentes relacionados a la lógica del juego. El juego no continua si no se revela una casilla, si o si se necesita ir revelando casillas para continuar, por ende y según lo mencionado anteriormente es que, el patrón observador nos entrega una solución general y un enfoque probado para resolver el diseño e implementación de nuestro software, además de mejorar su calidad.

5. Buenas Prácticas de Desarrollo

Durante el desarrollo del software hemos aplicado distintas prácticas para asegurar la calidad del mismo, entre ellas están; modularizar el proyecto, repartir la carga de trabajo entre los integrantes, pruebas exhaustivas de diversas situaciones posibles, incluir comentarios en el código para su correcta comprensión, revisiones en conjunto del código y los avances realizados, el respaldo constante del proyecto a medida que se iba avanzando, entre otras buenas prácticas.

6. Repositorio en GitHub

Se creó un repositorio en “GitHub” el cual posee un branche principal (main) y otros secundarios (1 por cada integrante), estos branche secundarios se hicieron para que cada integrante haga modificaciones en su propio branch y tenga su propia versión del proyecto, evitando así entorpecer el trabajo de los demás.

Antes de actualizar el branche principal se llegaba a un acuerdo y se analizaba lo mejor de cada versión de cada integrante.

Enlace Repositorio: <https://github.com/EduardoLealC/INFO229-ArquiSoftware>

7. Despliegue del Software

Protocolo de despliegue del software:

Nota: Sólo se requiere tener Python instalado en el sistema, aparte de esto no se requiere ninguna configuración adicional para ejecutar correctamente el programa.

- 1- Dirigirse al enlace GitHub y copiar el enlace del repositorio.
- 2- Crear una carpeta en tu sistema y abrir una terminal cmd en ésta.
- 3- Ejecutar el comando “git clone url” (la url es el enlace del repositorio).
- 4- Luego de clonar el repositorio, dirigirse mediante la terminal a la carpeta Principal ‘INFO229-ArquiSoftware’. (“cd INFO229-ArquiSoftware”)
- 5- Ahora dirigirse mediante la terminal a la carpeta ‘BuscaminasFinal’ (“cd BuscaminasFinal”)
- 6- Ya en esta carpeta, ejecutar el comando “python main.py” para correr el programa.

8. Conclusiones

La realización de este proyecto, mediante diversos métodos y buenas prácticas nos permitieron mejorar la calidad del software requerido y entender la importancia de estos mismos a la hora de desarrollar software.

El uso de la metodología 4+1 nos permitió tener una mejor comprensión de cómo funciona todo el sistema, y tener una documentación bien estructurada y entendible. El uso de un patrón de diseño nos facilitó la realización de todo el proyecto y además nos otorgó un orden a la hora de escribir el código y trabajar en equipo. Por último, el uso de buenas prácticas nos propone obligaciones necesarias y útiles a futuro, pensando en el mundo laboral real.

De esta manera podemos concluir que la aplicación de todo lo mencionado anteriormente a un proyecto de desarrollo de software, mejora considerablemente su calidad, comprensión y escalabilidad futura.